

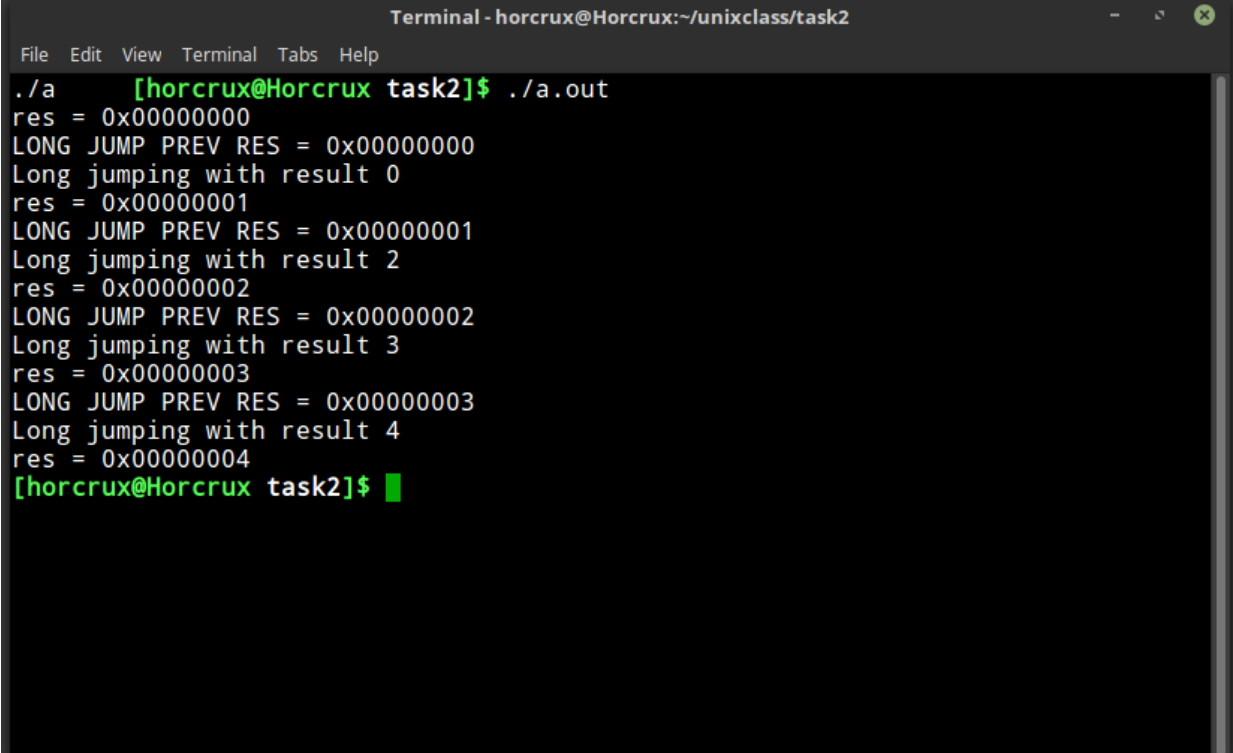
UNIX TASK – 2

Submitted By:
Rushab Shah
4nm15cs141

Prasanna Moolya
4nm15cs118

1. Setjump

```
#include <setjmp.h>
#include <stdio.h>
void testit (jmp_buf env, int prev_res){
    int res = (0 == prev_res) ? prev_res : prev_res + 1;
    printf ("LONG JUMP PREV RES = 0x%08x\n", prev_res);
    printf ("Long jumping with result %d\n", res);
    longjmp (env, res);
} /* testit () */
int main (int argc, char *argv[]){
    jmp_buf env;
    int res = setjmp (env);
    printf ("res = 0x%08x\n", res);
    if (res > 3)
        return 0;
    testit (env, res);
}
```

A terminal window titled "Terminal - horcrux@Horcrux: ~/unixclass/task2" showing the execution of a C program. The program uses setjmp and longjmp to demonstrate long jumps. The output shows the program running in a loop, printing the previous result and the new result after a long jump. The results are: res = 0x00000000, LONG JUMP PREV RES = 0x00000000, Long jumping with result 0; res = 0x00000001, LONG JUMP PREV RES = 0x00000001, Long jumping with result 2; res = 0x00000002, LONG JUMP PREV RES = 0x00000002, Long jumping with result 3; res = 0x00000003, LONG JUMP PREV RES = 0x00000003, Long jumping with result 4; res = 0x00000004. The prompt [horcrux@Horcrux task2]\$ is visible at the end of the output.

```
Terminal - horcrux@Horcrux: ~/unixclass/task2
File Edit View Terminal Tabs Help
./a [horcrux@Horcrux task2]$ ./a.out
res = 0x00000000
LONG JUMP PREV RES = 0x00000000
Long jumping with result 0
res = 0x00000001
LONG JUMP PREV RES = 0x00000001
Long jumping with result 2
res = 0x00000002
LONG JUMP PREV RES = 0x00000002
Long jumping with result 3
res = 0x00000003
LONG JUMP PREV RES = 0x00000003
Long jumping with result 4
res = 0x00000004
[horcrux@Horcrux task2]$
```

2. Race condition

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>

static void charatotime(char *);

int main(void)
{
    pid_t pid;

    if( (pid = fork()) < 0){
        perror("Fork error");
        exit(1);
    }

    else if( pid == 0){
        charatotime("Output from child\n");
    }

    else{
        charatotime("Output from parent\n");
    }

    exit(0);
}

static void
charatotime(char * str){
    char * ptr;
    int c;
    /* Ensure that characters sent to stdout are output as soon
       as possible - make stdout unbuffered. */
    setbuf(stdout,NULL);
    for(ptr = str; c = *ptr++; )
        putc(c, stdout);
}
```

File Edit View Terminal Tabs Help

```
[horcrux@Horcrux task2]$ gcc 2.c && ./a.out
```

```
Output from parent
```

```
Output from child
```

```
[horcrux@Horcrux task2]$ gcc 2.c && ./a.out
```

```
Output from parent
```

```
Output from child
```

```
[horcrux@Horcrux task2]$ ./a.out
```

```
Output from parent
```

```
Output from child
```

```
[horcrux@Horcrux task2]$ gcc 2.c
```

```
[horcrux@Horcrux task2]$ ./a.out
```

```
Output from parent
```

```
Output from child
```

```
[horcrux@Horcrux task2]$ ./a.out
```

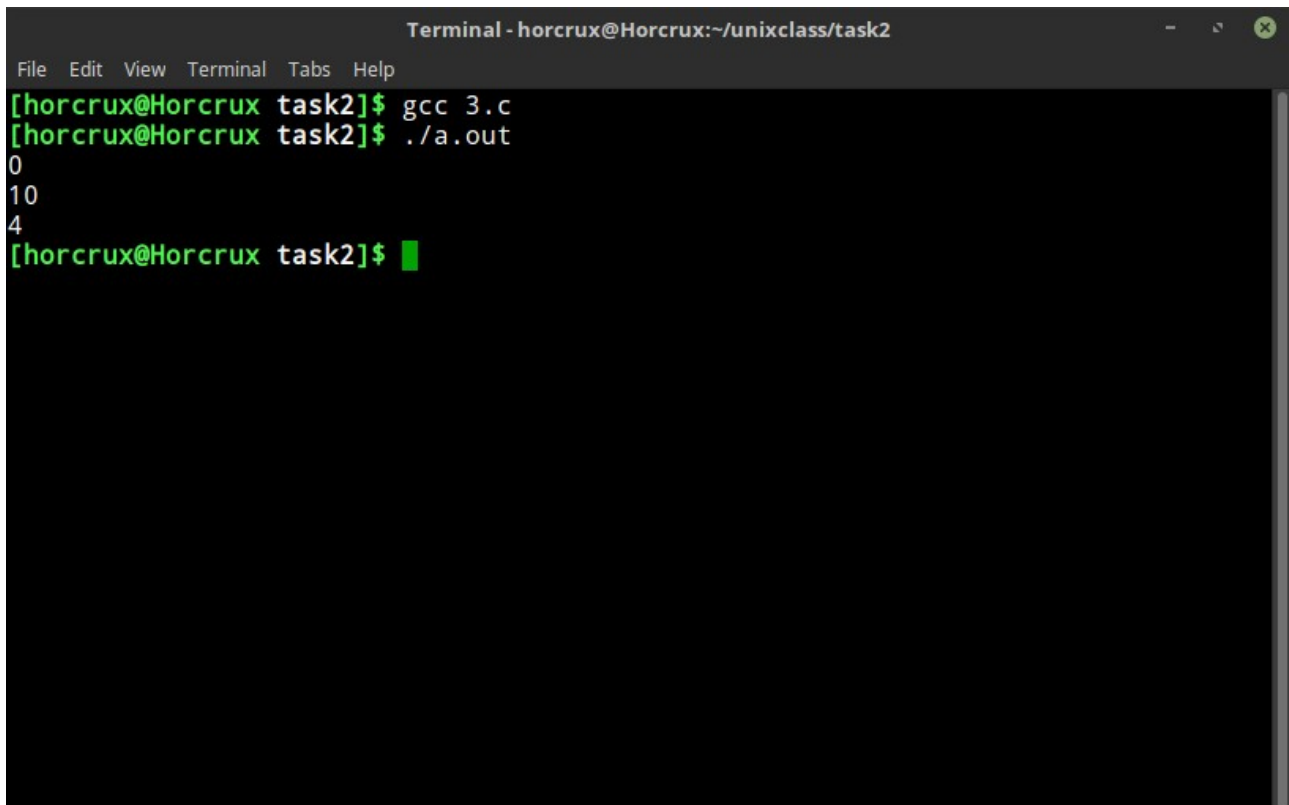
```
Output from parent0
```

```
utput from child
```

```
[horcrux@Horcrux task2]$ █
```

3. Sigconn

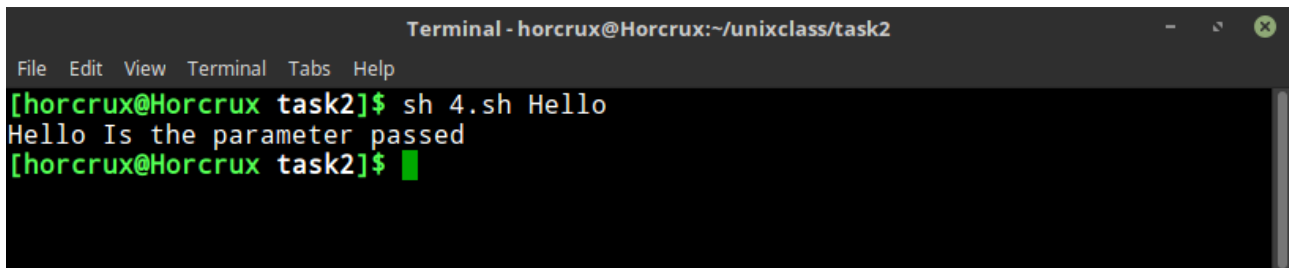
```
#include<stdio.h>
#include<setjmp.h>
#include<signal.h>
sigjmp_buf env;
void print(int a){
    printf("%d\n",a);
    siglongjmp(env,4);
}
int main(int argc,char* argv[]){
    int res = 0;
    if((res = sigsetjmp(env,1)) == 0){
        printf("%d\n",res);
        print(10);
    }
    printf("%d\n",res);
    return 0;
}
```

A terminal window titled "Terminal - horcrux@Horcrux:~/unixclass/task2" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [horcrux@Horcrux task2]\$. The user enters gcc 3.c, then ./a.out. The program outputs 0, 10, and 4 on separate lines. The prompt returns to [horcrux@Horcrux task2]\$.

```
Terminal - horcrux@Horcrux:~/unixclass/task2
File Edit View Terminal Tabs Help
[horcrux@Horcrux task2]$ gcc 3.c
[horcrux@Horcrux task2]$ ./a.out
0
10
4
[horcrux@Horcrux task2]$
```

4. Interpreter

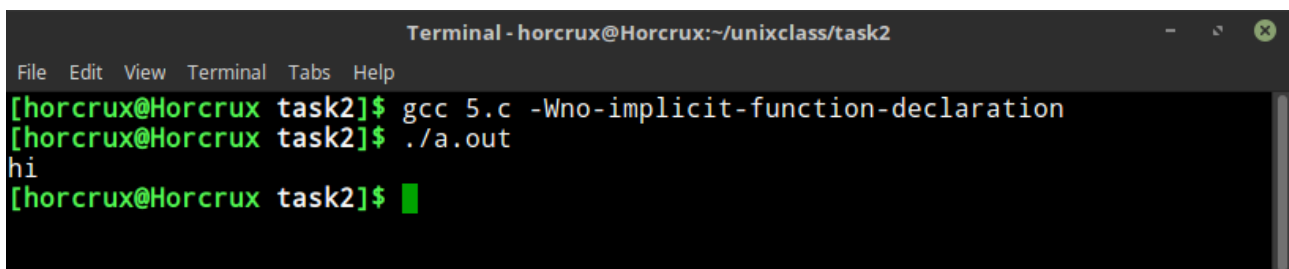
```
#!/bin/sh  
echo "$1 Is the parameter passed"
```



A terminal window titled "Terminal - horcrux@Horcrux:~/unixclass/task2" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [horcrux@Horcrux task2]\$. The user enters "sh 4.sh Hello". The output is "Hello Is the parameter passed". The prompt returns to [horcrux@Horcrux task2]\$. The terminal has a dark background with green text.

5. System Function

```
#include <stdio.h>  
int main(){  
    system("echo hi");  
}
```



A terminal window titled "Terminal - horcrux@Horcrux:~/unixclass/task2" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [horcrux@Horcrux task2]\$. The user enters "gcc 5.c -Wno-implicit-function-declaration". The prompt returns to [horcrux@Horcrux task2]\$. The user enters "./a.out". The output is "hi". The prompt returns to [horcrux@Horcrux task2]\$. The terminal has a dark background with green text.